

THE ROLE OF XML IN THE MODELING PROCESS OF A VIRTUAL BUSINESS

Lucia Rusu¹
Cristina Nemeş²
Marius Podean³

Abstract

The aim of this paper is to describe the XML stack of languages used in the implementation process of a web application. This application is based on a three tier architecture named XRX. In this type of architecture there is no need for data model transformations between the tiers of the architecture like in the classical architecture. So the applications developed according XRX architecture become more flexible, efficient and simple.

Keywords: XML, XPath, XQuery, XSLT, XForms, XRX, UBL

1. Introduction

Due to the expansion of the Internet every day new sites and web applications appear. To stay competitive on the virtual market these applications must constantly improve the way they exchange data.

Web applications are generally based on a three-tier architecture that uses HTML or XHTML in the browser. At the logical level is used PHP, Python or Perl or for more complex applications object-oriented languages like C # or Java. At the database tier is used the well known SQL.

The objective of this paper is to present that a web application can be developed using only Web standards by implementing a three-tier architecture that uses in a) browser XForms and XHTML, b) Representation State Transfer (REST) interfaces, and c) at the database level the query language named XQuery. After [Kaufmann and Kossmann, 2009] applications developed according to XForms-Rest-XQuery (XRX) architecture become more agile, flexible and simple.

To achieve these objectives we present a study of published works about XRX architecture and then about XML and XML family languages such as XPath, XQuery, XSLT.

¹ Lucia Rusu is Professor at Babeş-Bolyai University of Cluj Napoca, Business Information Systems Department, email: lucia.rusu@econ.ubbcluj.ro

² Cristina Nemeş is E-Business master at Babeş-Bolyai University of Cluj Napoca, Business Information Systems Department, email: nemes.andreeacristina@gmail.com

³ Marius Podean is PhD. at Babeş-Bolyai University of Cluj Napoca, Business Information Systems Department, email: marius.podean@econ.ubbcluj.ro

A new idea for developing and implementing Web applications is based on W3C standards which includes data format (XML, HTML, XHTML), programming languages (XQuery, XPath, XSLT), schema (XML Schema) and protocols (HTTP, REST) [Kaufmann and Kossmann, 2009].

If data in XML format is stored in the browser in XForms format and in a native XML database, the application is very agile thanks to XRX architecture [McCreary, 2010b].

In section 2 we provide some major features about XML family. In section 3 is described the XRX architecture and in section 4 are presented some conclusions and future work.

2. XML Family

2.1. XML

XML is used for describing data and its structure, so in XML documents we can distinct between content and form. XML has the ability to separate structure from format. In XML we can define our own tags because there are no predefined tags, so XML is characterized by extensibility.

XML has limitations on recursive structures and polymorphism is the only object oriented principle respected. After [Abrevaya, 2003] XML describes data in order to be both hard and soft independent. XML is the standard used for data exchange.

XML is a simple and standard way to interchange text data in a structured format between different programs. XML separates data from presentation and is used for transmitting data between applications. Another benefit of XML is that it unifies the universe of documents with the universe of data providing a single mode to present its structure [Harold and Means, 2004].

In recent years XML has become very important in data exchange which resulted in the development of relational databases with XML support and native XML databases.

Many software producers have realized the importance of using XML to transport data and had developed native XML databases or servers with support for XML. Such servers are: eXist, MarkLogic XML Server 4.0, IBM DB2 PureXML Server, Oracle Berkeley DB XML has support for XML, XSLT and XQuery.

Using XML documents to transmit data between applications and between its tiers, applications become more extensible and flexible, with the ability to be integrated with Web applications running on different platforms and using different technologies. XML is a language used for easy exchange of structured documents through the Internet.

In the syntax of XML language metadata are attached using specialized tags. XML is considered a metalanguage because it is used to create other languages called vocabularies.

XML [Harold and Means, 2004] was originally created as a subset of the SGML language for easy transfer between browsers. Over time XML has become a standard language for sending data between computers or different companies who need to work with common data but do not have the same configuration for the data processing systems.

The most used XML languages are XQuery, XPath, XSLT, XML Schema that although have different purposes they complement each other.

2.2. XPath

Many XML languages such as XQuery, XSLT use path expressions to navigate in the hierarchical structure of an XML document which is modeled as an ordered tree [Kay, 2004]. A path expression locates a node of the tree, so the result is a sequence of distinct nodes in the order they appear in the document. The sequence of nodes can be filtered by a predicate expression contained between “[]”. Predicates are evaluated for each node in the sequence of nodes, and if returns "true" they are displayed otherwise are not shown. Predicates are used to limit the extracted data from XML documents.

XPath role as support for other languages [Buchmann, 2007] :

- XQuery - used for query processing;
- XSLT - XML nodes are transformed into other XML nodes;
- XPointer - makes references to the nodes addressed by XPath expressions;
- XML Signature and XML Encryption - addresses nodes for signature and encryption.

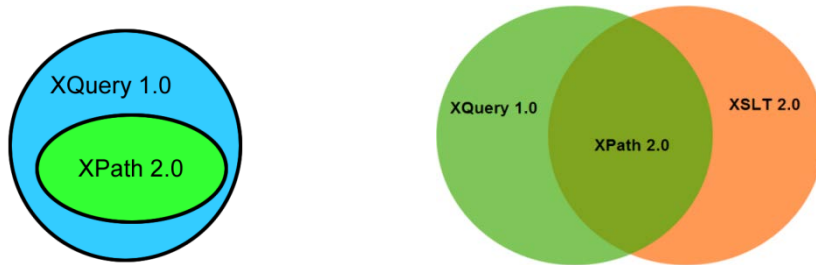
XPath addresses a document in the same way in which are addressed the files path in a directory structure [Arba, 2008]. XPath language is used by many XML languages to locate and access information from an XML tree.

2.3. XQuery

XQuery [Boag et al., 2010] is the language used to query XML data and is built on XPath expressions. XQuery is a query language, an extension of the XPath language which allows to select information according to one or more criteria, joining data from multiple documents, mathematical operations on numeric data type or date type [Walmsley, 2007]. XQuery extends XPath by introducing the FLWOR expressions (FLWOR expressions are the terms: for, let, where, order by, return), predefined functions and user-defined functions.

XQuery language extends XPath language, and like XSLT uses XPath to reference nodes through paths. (Figure 1)

Figure 1. Relations between XQuery-XPath-XSLT



Source: [Heitmeyer, 2009]

Any XQuery query contains two phases [Buchmann, 2007] :

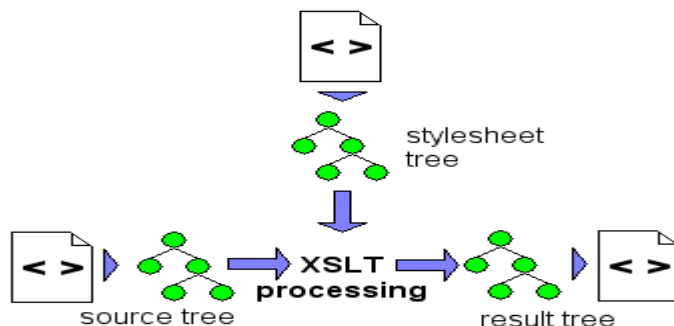
- analyze phase - reads XML fragments;
- execution phase - returns a result.

XQuery is not a XML vocabulary, but there is also a XML version of it called XQueryX. XQuery has the same role that SQL has in relational databases, and such as SQL query result can be saved as a relational table through XQuery we can create XML documents.

2.4. XSLT

eXtensible Stylesheet Language (XSL) [Anders, 2006] was created by World Wide Web Consortium and was initially designed for formatting XML documents. It was divided into: XPath to locate nodes, presentation format (XSL-FO / XSL - Formatting Objects) and an XML transformation language (XSLT or XSL-Transformations) [Anders, 2006]. XSLT can make transformations from XML into HTML or text formats, so a general definition can be: XSLT is a language for transforming the structure of XML documents.

Figure 2. XSLT processing



Source: [Dubinko, 2003]

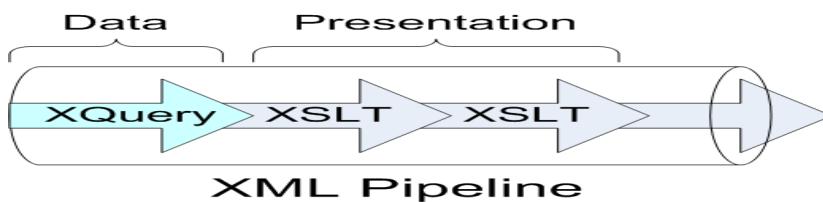
XSLT works with trees resulting from the parsing process of XML documents. To make XSLT transformations we need XML documents which are going to be transformed and a stylesheet that contains transformation rules. These documents are represented as trees in memory. After the XSLT processing results another tree that will be another XML or HTML document. (Figure 2)

XSLT transforms XML documents into a tree structure, so XSLT works with the tree structure and not with the document itself. Each component of the XML document (elements, attributes, processing instructions) are treated as nodes of a tree. XSLT is a language that allows selecting nodes and conducting certain operations on certain nodes [Ducharme, 2001].

Thereby XSLT is used to transform an XML document into another XML document with a different structure, and XSL-FO is used for formatting and styling. Transformation involves rearranging, renaming, adding, deleting and extracting tags from an XML document and styling involves the transformation of an XML document for formatting a document such as PDF in order to achieve a presentation [Buchmann, 2007].

XSLT has XML syntax but XQuery has a non-XML syntax. XQuery is used for queries on collections of documents, and XSLT is used to transform XML documents into other XML or HTML documents. XQuery is used for processing data from multiple sources while XSLT is more suitable for transforming one data model into another. Thereby XQuery is recommended to be used when working with data and XSLT for presentations (Figure 3).

Figure 3. Comparison between XQuery - XSLT



Source: [Heitmeyer, 2009]

In general XML data is transformed in order to be displayed or published like HTML and PDF. The most often used transformation transforms XML documents in HTML documents.

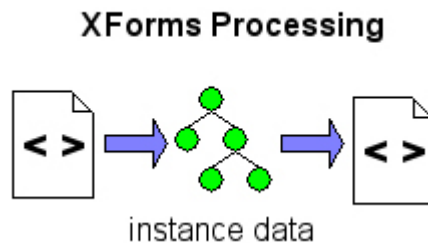
2.5. XForms

W3C has developed a new platform-independent markup language for developing Web forms with richer interactions, called XForms [Boyer, 2007], which removes the need to use scripts and provides the opportunity to achieve validation, calculations and setting

data types. These new features offered by XForms can be made without JavaScript or AJAX.

Similarly to the processing of XSLT, in XForms processing an instance or an XML document stored on a server are parsed in memory. At submit the instance of data is sent to the server in XML format. (Figure 4)

Figure 4. XForms processing



Source: [Dubinko, 2003]

XForms communicates with the server through XML documents, which allows displaying documents with dynamic interfaces. A browser receives an XML document that will be displayed according to the template then the user can edit the document and send it to the server.

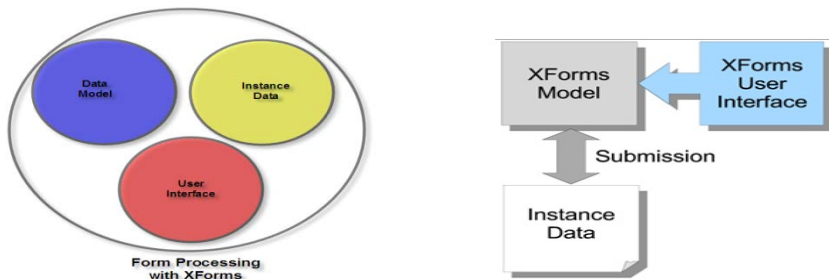
On the Internet through forms are created search engines, surveys, e-commerce so that most interactions with the client on the Web is done using forms. Among the disadvantages of traditional HTML forms are: very poor integration with XML, combining presentation with content, dependence on JavaScript and Ajax for validation.

XForms aims to combine XML with forms and has the following features: full support for XML, separation between presentation and content, making validation without using JavaScript and Ajax code. Because forms are used to collect and work with data, the essential term in XForms is "instance data" which is presented in XML format.

XForms divides classic XHTML forms in three parts: instance data, model and user interface created by the controls. Because of this separation XForms allows separation between presentation and content, making calculations and validations.

A form in addition to the main function to collect data has to perform calculations, validations, to determine which fields are required and read-only. HTML forms make these processes directly by its scripts. XForms aims to achieve this functionality without using scripts by using the features offered by XPath (calculations, validations) and XML Schema (data type, constraints) [Boyer, 2007].

Figure 5. XForms components



Source: [Heitmeyer, 2009] & [Jaghoori, 2010]

The instance is an XML document and must have the “xmlns” attribute even if it is empty. Binds are used to perform validations, calculations, constraints and to determine the data type. Binds refer a single marker in the instance through an XPath expression. Instances and binds form the model's form. Controls are used to present information. They appear in the form in the order they have been defined. (Figure 5)

XForms allows the construction of forms in a very similar way to HTML. To use XForms it is necessary to declare the data structure containing information that needs to be displayed or modified. Thus XForms works with data stored in XML format. XForms is modeled on MVC (Model-View-Controller) model so the “model” represents the data structure in XML format, and "view" represents the controls on the form.

In classic web applications JavaScript is used in order to achieve validation, dynamic communication with the server, constraints and calculations. It can be used both client-side and server-side. This language has a similar syntax to C language and for scripts implementations it is necessary to create functions.

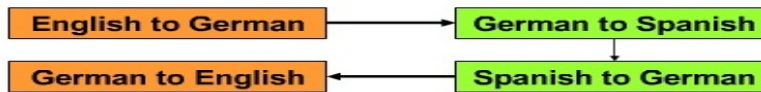
An alternative to JavaScript is XForms that although is not an object oriented language can successfully replace the scripting language. XForms can implement the same requirements without using JavaScript. For calculations, validation, dynamic communication with the server is not required knowledge about object oriented languages. XForms makes these operations in a very simple way through binds.

3. XRX

XForms-REST-XQuery (XRX) uses for client-side XForms, REST and XQuery on the server. XRX is

a hybrid architecture that combines three different but compatible standards. To understand the XRX architecture we can perform the following example: consider a text in English to translate it into German using Google Translate, and then we will translate German into Spanish. Then we translate the text from Spanish into German, and finally German to English. We see that the result is not the same as the initial text. (Figure 6)

Figure 6. Translations English-German-Spanish



Source: [McCreary, 2008]

The benefits of the XRX architecture [McCreary, 2010a]:

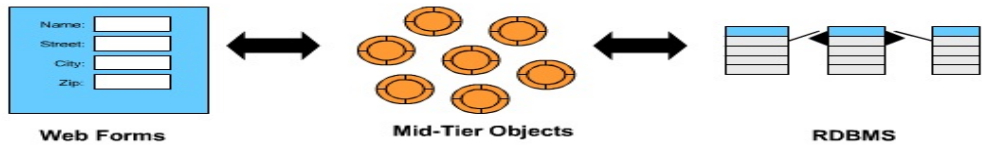
- simplicity - because data is represented in the same way on the client and server there is no need for transformations from one format to another. Applications based on traditional HTML forms saves data in key-value pairs in Web browsers, in hierarchies of objects in the middle layer and relational structures. In this case it is necessary to make translations from one format to another;
- rapid development of prototypes - because there is no need for translation between client, middle tier and server, XRX applications can be used as a rapid development medium;
- agility - XForms elements can be dynamically generated;
- consistent - because XRX elements are derived from a central model, the form may be more consistent. If certain rules are modified in a model, then all forms that use those rules can be easily modified.
- compatible with W3C standards: CSS, XML Schema, XPath, XForms, XQuery.

If the application is designed according a three tier architecture and is based on object-oriented languages such as C # and Java the key-value pairs from HTML must be converted into object-oriented language. Then these objects are converted into tabular data streams to be stored in relational databases. If the user wants to see some data from the database, this data must be transformed into objects, then in key-value pairs to be displayed.

In a classical architecture on three levels are needed four translations between the levels (and if it is necessary to use XML Web services we add two more translations: objects in XML and XML into objects) [McCreary, 2010b]:

1. HTML in objects;
2. Objects in RDBMS;
- 3.RDBMS in objects;
4. Objects in HTML.

Figure 7. Classical three tier architecture



Source: [McCreary, 2008]

Representation State Transfer (REST) [Fielding, 2000] is a group of constraints above the client-server architecture. REST principles include requests, resources, identifiable URLs, unique and standardized interfaces. Information is represented as resources that are identified by a URL and can be represented in several ways (PDF / HTML / XML). In recent years, developers are using Web features and create Web services based on REST services instead of using WSDL / SOAP.

REST is a Web architecture that provides principles to identify errors in applications. REST contains a set of architectural constraints used to minimize latency and network communication. REST allows capture and reuse of interactions, dynamic replacement of components and processing activities by intermediaries.

4. Conclusions

As we presented above the XML stack of languages is very well suited with different languages. Each of these languages has a different functionality in order to allow the possibility to develop web applications using only XML vocabularies.

XML is a language that separates content from presentation and is used especially on the Internet to exchange data between different web applications. XPath is used in many languages like XSLT, XQuery to address and locate nodes. XQuery and XSLT cannot be compared because they have been created for different purposes. Thus, XQuery is used for working with data, and XSLT is used to transform XML documents in HTML, text or other XML documents. XForms represents the next generation of HTML forms, and has new features like the possibility to validate data without using JavaScript.

All these languages should be used together in order to complete each other and develop complex web applications. The applications developed using these languages and the XRX architecture are more simple, efficient and flexible. Using these languages in the future will surely improve the way in which the companies exchange data.

Acknowledgment

This paper was supported by Romanian National Authority for Scientific Research under the grant no. PN2 92-100/2008 SICOMAP

References

[Abrevaya, 2003] Abrevaya, A. (2003). Web Service Aggregation and Reporting with XQuery. <http://www.devx.com/assets/devxads/6491.pdf>.

[Anders, 2006] Anders, B. (2006). Extensible Stylesheet Language (XSL) Version 1.1. W3C Recommendation 05 December 2006 <http://www.w3.org/TR/xsl11/>.

[Arba, 2008] Arba, R. (2008). Contribuții la conceperea, proiectarea și implementarea unui sistem de comerț electronic ca suport pentru afacerile firmelor românești. PhD thesis, Universitatea Babeș-Bolyai Facultatea de Științe Economice și Gestiunea Afacerilor Cluj-Napoca.

[Boag et al., 2010] Boag S., Chamberlin D., Fernandez M., Florescu D., Robie J., Simeon J. (2010). XQuery 1.0: An XML Query Language (Second Edition). W3C Recommendation 14 December 2010 <http://www.w3.org/TR/xquery/>.

[Buchmann, 2004] Buchmann, R. (2004). Contribuții la conceperea, proiectarea și realizarea afacerilor pe Internet. PhD thesis, Universitatea Babeș-Bolyai Facultatea de Științe Economice și Gestiunea Afacerilor Cluj-Napoca.

[Buchmann, 2007] Buchmann, R. (2007). Rolul XML în interoperabilitatea sistemelor informatice pentru afaceri. Editura Risoprint, Cluj-Napoca.

[Dubinko, 2002] Dubinko, M. (2002). What Are XForms. <http://www.xml.com/pub/a/2001/09/05/xforms.html?page=1>.

[Dubinko, 2003] Dubinko, M. (2003). XForms Essentials. O'Reilly, Sebastopol, CA, First edition.

[DuCharme, 2001] DuCharme, B. (2001). XSLT Quickly. Editura Manning, Greenwich.

[Fielding, 2000] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

[Harold and Means, 2004] Harold, E. R. and Means W. S. (2004). XML in a Nutshell, Third Edition. Editura O'Reilly Media

[Heitmeyer, 2009] Heitmeyer, D. (2009). CSCI E-153, Web Development Using XML. http://cscie153.dce.harvard.edu/lecture_notes/2009/20091110/handout.html

[Jaghooori, 2010] Jaghooori, B. (2010). Why XForms might be a Winner. <http://blog.mgmt-tp.com/2010/03/rethinking-web-forms-xforms-part1/>.

[Kaufmann and Kossmann, 2009] Kaufmann, M. and Kossmann, D. (2009). Developing an Enterprise Web Application in XQuery. In International Conference on Web Engineering, pages 465–468.

[Kay, 2001] Kay, M. (2001). XSLT: Programmers Reference. Editura Wrox.

[Kay, 2004] Kay, M. (2004). XPath 2.0 Programmers Reference. Editura Wrox.

[McCreary, 2008] McCreary, D. (2008b). XRX Training Classes. <http://www.danmccreary.com/training/xrx/index.html>.

[McCreary, 2010a] McCreary, D. (2010a). Syntactica Solutions for XRX Developers. <http://www.syntactica.com/solutions/xrx.xq>.

[McCreary, 2010b] McCreary, D. (2010b). Using Native XML Systems to Manage Metadata. <http://www.tdan.com/view-articles/14517>.

[Walmsley, 2007] Walmsley, P. (2007). XQuery. O'Reilly Media, Inc.

[Williams, 2009] Williams I. (2009). Beginning XSLT and XPath. Editura Wrox.

[Wolfgang, 2009] Wolfgang, M. (2009). eXist-db Open Source Native XML Database. <http://exist.sourceforge.net/credits.html>.